DOCUMENT RESUME

ED 055 471 FL 002 310

AUTHOR Milic, Louis T.
TITLE The Possible Usefulness of Poetry Generation.
PUB DATE 70
NOTE 19p.

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Programs; Computers; Form Classes (Languages); Generative Grammar; Grammar; Haiku; Imagery; *Language Research; Language Rhythm; Literary Conventions; *Metaphors; Phonology; *Poetry; Semantics; Sentences; Syntax; Taxonomy; Transformation Generative Grammar; *Versification

ABSTRACT

The possible usefulness of computer poetry is concerned with what the programmer can learn about language, about poetry, and about poets. The problems in designing computer programs to construct poetry include considerations in generating well-formed sentences which have the added restrictions that poetry requires: meter, rhyme, logic, diction, subject matter, and imagery. These restrictions are phonological and semantic. At the present stage of development of semantic theory, such linguistic features as imagery are of minor interest for poetry generation. The basic difficulty in poetry generation is introducing enough uncertainty to insure variety in the resultant output without violating the constraints that govern well-formed sentences. Numerous linguistic restrictions can be added to computer programs to refine the poetic output. Generating computer poetry underlines the curious behavior of familiar words in unfamiliar combinations. In poetry, metaphor is readily accepted as an alternative to calling a sentence nonsensical. Metaphor is thought to be the special creation of the poet; this presents a curious problem when the poet is a computer. (Author/VM)

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

THE POSSIBLE USEFULNESS OF POETRY GENERATION

Louis T. Milic

The Cleveland State University

"PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY Louis T. Milic

TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE U.S. OFFICE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER."

ED055471

One interesting result of my activity in computer poetry generation is a new definition of poetry. In an important sense, strings of words are interpreted as poetry if they violate two of the usual constraints of prose, logical sequence and semantic distribution categories. Such a sentence as "The pond under my rocker saw the sunrise so the porch post planted carefully" exhibits both of these. Because ponds cannot see and porch posts cannot plant the semantic constraints of English would normally prohibit their being used as subjects of verbs displaying human (or animate) action. And because even if a pond could see the sunrise that would provide no satisfactory justification, as indicated by the conjunction so, for the activity of the porch post, we conclude that the usual logical requirements of discursive prose have also been violated. In short, since the sentence is obviously well-formed syntactically but does not "make sense," it is interpreted as poetry, as part at least of a larger poetic structure. This conclusion does not seem whimsical. Rather it seems to shed some light both on how we read poetry of our own time and how poets operate. It might not be excessive to say that such a conclusion justifies what many consider a rather foolish pastime, computer poetry generation.

It might be pointed out, of course, that in the Western world, the writing of poetry has always been considered a wasteful thing for an able-bodied man to do, who could be earning a living in a more serious way. The purpose

FL002 310

© Copyright Louis T. Milic, 1970

of computer poetry generation differs, however, from the normal poet's activity in that the usual poetic intention (prophecy, self-expression, aesthetic creation) is replaced by a heuristic one. The possible usefulness of computer poetry is concerned with what the doer can learn about language, about poetry and about poets from this sort of simulation. If his product has poetic interest so much the better but he has no illusions that he will be set beside Pope, Shelley and the other inhabitants of the Parnassus. In this, his aim is more modest than that of the computer musicians, who have some respectable compositions to their credit, and the producers of computer graphics, whose beautiful arrangements of lines adorn a number of walls in good artistic company. Apparently, music art and poetry, whatever they may have in common, are significantly differentiated both in their elements and in our expectations of how these combine into aesthetic structures.

The random generation of sentences is a fairly simple procedure, especially when programmed in a high-level computer language like SNOBOL 4.* A formula for a sentence is established, say the following:

Article/Noun(1)/Verb/Preposition/Article/Noun(2).

A list is provided for each word-class. For example, the list in Noun(1) is: boy, girl, shirt, cat, table, book, house, knife, box. The program instructs the computer to select the first word in each appropriate list, to reconstitute the list with the word just used at the end, and to print the resultant sentence. Provided the lists are of different lengths, the program will generate by this

---

* It ought to be noted that SNOBOL4 is quite inefficient compared with FORTRAN and requires a good deal more computer time as well as approximately 200k bytes of core, but it is easy to learn and less critical of errors than more efficient languages.

means a great number of different sentences of the following type:

3. A girl picked about a script

67. A shirt thought with a dog

78. The table thought of the window

88. The book sat around the dog

93. A girl learned about a man

104. The cat learned in the hair

107. A house imitated to a man

335. A boy learned to a hair

341. A house sat to a picture

392. The cat studied in the curtain

Syntactically, these sentences are well-formed in that the order of the word-classes is consistent with English grammar and the inflection is properly managed. Many of the sentences would obviously be considered perfectly good English by a native speaker (e.g., 93). Many others, however, though syntactically adequate would be judged somehow defective. Some of those cited above (67, 78, 107), for example, have inanimate subjects with verbs that require human or animate ones. Other defects result from the tendency of certain collocations to acquire idiomatic meaning: "to a man" (107), "sat to" (341), "to a hair" (335). These unpredictable results suggest that a grammar for generating sentences should be linked to a dictionary so as to prevent the occurrence of aberrant sentences. Several sentences, however, are not clearly well-formed or defective. In sentences 88 and 392, a little stretching on the interpretive level permits us to accept the notion of a book sitting around or a cat studying. In the proper context such sentences would go completely

unquestioned. By context is meant not only the verbal environment (i.e., the sentences before and after) but the situational context (i.e., whether the sentence is identified as poetry). What this suggests is the difficulty of producing a grammar of poetical sentences: if the grammar produces only well-formed sentences it inhibits certain metaphorical possibilities. (Reference 6, passim).

The number of different sentences which can be generated with a short vocabulary, though not infinite, is quite astronomical if a recursive feature is introduced. If the sentences are not built to a simple pattern like the one shown above but permitted to be short or long according to some device built into the program the results can be extremely varied. A linguist took the first ten sentences of a children's story and derived from it 77 rules which recursively could produce 10^{20} different sentences, of thich these are a selection:

Engineer Small is polished

When he is polished, he is proud of smokestacks and fire-box.

Steam is shiny.

The water under the wheels is oiled whistles and its polished, shiny, and big trains is black.

Because of the necessary repetition, most of the sentences would not be very interesting.

A more interesting sentence produced by Victor Yngve of MIT is "What does she put four whistles beside heated rugs for?" (14) For some obscure reason this sentence has a great deal of interest for I have seen it quoted often since I first mentioned in a talk at IBM in 1964. In fact, one poet felt obligated to write a poem in reply in which he incorporated the line.

Despite this unusual response, these utterances are not presented as poetry, merely as randomly-generated sentences.

Whatever rules are established for the production of sentences, the rules for producing poetry are generally more complicated because they involve the selection of the sentences according to more stringent rules. Obviously, poetry does not usually consist merely of stringing together any sentences that may happen to occur. In the production of poetry, sentences are produced in conformity with further constraints, such as meter, rhyme, logic, diction, subject-matter, imagery ... Pope's line, "The hungry judges soon the sentence sign," is followed by another line to the effect that the accused are denied justice because of the impatience of the officials of the court. Let us rewrite the couplet:

The hungry judges soon the sentence sign
And the accused are railroaded as usual.

Obviously this formulation lacks something of being a perfect Augustan couplet. After the application of a series of further constraints, we might get something like Pope's second line,

And wretches hang that jurymen may dine.

As everyone knows, poetry is more difficult to write than prose.

But there is one sense in which poetry is freer than prose and that is the willingness of the reader to interpret a poet, no matter how obscure, until he has achieved a satisfactory understanding. Modern poets have exploited this willingness. One of the more obscure poets of great renown is the Welshman, Dylan Thomas, whose fame in the United States is a little puzzling and cannot wholly be ascribed to the drinking habits he displayed

during his journey there. A cursory glance at his poetry reveals lines such as these, which are not immediately grasped by the average reader:

> Because the pleasure-bird whistles after the hot wires,
>
> Shall the blind horse sing sweeter?

According to one of his childhood friends, Thomas constructed some of his poems in what is surely an unorthodox way. Thomas's aim

> was to create pure word-patterns, poems depending upon the sounds of the words and the effect made by words in unusual juxtaposition.... he would draft the general scheme of a stanza, leaving some of the words to be filled in later. He carried with him a small notebook containing a medley of quite ordinary words, most of them very short--tree, bough, hive, gold, numb, and so on. When he wanted to fill in a blank he read in his dictionary, as he called it, and tried one word after another, so that he would obtain (let us say) tree of night, bough of night, and hive of night. An unusable arrangement often suggested other possibilities; thus, the rejected hive of night might be replaced by starry hive.
>
> If nothing in his dictionary satisfied him, he would sit with his mouth partly open, hoping to pick up a promising word from someone's conversation. As a last resort he might open a book at random and experiment with any likely word in the top line of each successive page.

This procedure is plainly reminiscent of a sentence-generating program. It is interesting to reconstitute the poet's private dictionary and to derive from it poetical fragments which he never got around to trying, or perhaps rejected. The creation of new Dylan Thomas poems on this basis is not terribly difficult and very instructive. Here is a poem which I have selected for this experiment. It is titled "In the Beginning," and consists of five stanzas of six lines, each one beginning with the title phrase. Here is the first stanza:

> In the beginning was the three-pointed star,
> One smile of light across the empty face;
> One bough of bone across the rooting air,
> The substance forked that narrowed the first sun;
> And, burning ciphers on the round of space,
> Heaven and hell mixed as they spun.

Careful reading will show the traces of the method described by Thomas's friend: "bough of bone," "minted face," "blunt flower."

My procedure consists very simply of taking out all the nouns in the poem, alphabetizing them and re-inserting them in the poem in this new order. The resultant form adds the element of alliteration to the other poetic devices of the poem. Here is the result:

> In the beginning was the three-pointed air
> One bough of bone across the empty blood
> One basis of bases across the rooting breath
> The birth forked that narrowed the first brain
> And burning brain on the blood of ciphers
> Crosstree and cloud mixed as they spun.

In the second stage, the verbs are re-arranged in the same way and the new version now reads as follows:

> In the beginning was the three-pointed air
> One bough of bone across the empty blood
> One basis of bases across the rooting breath
> The birth forked that abstracted the first brain
> And burning brain on the blood of ciphers
> Crosstree and cloud mixed as they burst.

The final permutation includes the adjectives, as well as the nouns and verbs. The result reads this way:

> In the beginning was the blunt air
> One bough of bone across the cloudy blood
> One basis of bases across the empty breath
> The birth forked that abstracted the first brain
> And burning brain on the blood of ciphers
> Crosstree and cloud first as they burst.

I do not claim that this effort to place words in even more unusual juxtaposition than Thomas managed has any poetical value, though some of the lines are not without poetic interest. With some refinement of the procedure, the result could be made much more attractive and consistent, perhaps sufficiently so to pass for a genuine example of the poet's work.

A more ambitious effort involved the creation of a set of quatrains distantly modelled on the first stanza of Blake's "A Poison Tree":

I was angry with my friend:
I told my wrath, my wrath did end.
I was angry with my foe:
I told it not, my wrath did grow.

As compared with the Dylan Thomas effort, which simply adopts the form of the original poem, the pseudo-Blake is of a different kind. It has a static form, in that every stanza has the same number of words as every other. Its peculiar features are the rhymes and the attempt to achieve consistency of internal reference. The following example is typical:

1. You were happy with your foe
   And your puny joy did glow
   Your foe relived your joy
   And you renewed the foolish ploy.

I will not claim a great deal of poetic merit for this quatrain but it is possible to point out that it rhymes, it scans in an approximate way and, most difficult of all, it has a certain consistency. The rhyming is achieved by arranging the line-ending words in separate lists. Whichever list is randomly selected for the first line-ending word, the next line-ending word must come from the same list. The same occurs for the second pair of lines. Because the lists are arranged in fixed arrays the same number of rhyming words must occur in each list. In English this can be a considerable problem, especially for words like gaiety. Ploy (in line 4) represents the exhaustion of my resources for words to rhyme with joy. The scansion is controlled by establishing a formula which scans and using words in each position of the same syllabic length and status pattern. The internal consistency is maintained by relating all pronouns and pronominal adjectives to the first word in line 2. Similarly the choice of the adjective happy in line 1 determines

the noun joy in lines 2 and 3, whereas the noun ending line 1 appears as the first noun in line 3. Unfortunately, the constraints severely limit the possible variety of the output. As soon as the first adjective is selected for line 1, the rhyme-scheme of the second couplet is determined. By means of a relatively simple system of pre-determined relationships of this type, something approximating sense can be produced over a span of four lines consisting of two sentences. Over a larger span, the problem of maintaining sense without falling into tautology would be progressively greater except by means of a highly sophisticated semantic compatibility procedure.

On the semantic level an interesting by-product of this set of permutations is the finding (also observed by Yngve, 14, p. 71) that words change meaning drastically in contexts which may be only slightly different. Such words as gay, bride, puny in collocation with each other develop unexpected possibilities:

143. You were gay with your bride
But your great gaiety did hide.
Your bride renewed your gaiety
And you reviewed the foolish satiety.

34. He was grieved at his bride
But his puny grief did chide.
His bride relieved his grief
And he renewed the silly belief.

Syntactically, it is also notable that line 2 can be regularly read as an inversion (i.e., the object before the verb).

In a static format of the type just described a number of ingenious solutions to the problem of variety are possible. An initial subordinate clause arrangement was tried by some experimenters at the RCA laboratories (5) in Lancaster, Pennsylvania, and produced the following stanza among a

thousand:

27. While life reached evilly through empty faces
While space flowed slowly o'er idle bodies
And stars flowed evilly upon vast men
No passion smiled.

Randomness was achieved by manual intervention, the operator depressing an interrupt button to select a word for each position. At Michigan State University, John Morris produced a number of haiku-type poems, using a vocabulary derived from an anthology of Japanese haiku (10). The basic constraint, aside from a plausible syntax, consisted of the requirement that seventeen syllables be distributed over three lines. There were no semantic constraints. The results are like these, apparently the best of four thousand produced by the Michigan computer:

Still midnight, silent,
Still waters, still frozen,
Battle dusk, and far.

The savage, savage
Scarecrow, down in silent dusk,
Frozen, well frozen.

Distance, I listen:
Far weird savage frozen spring,
Old song, echo still.

Glittering midnight:
Our hollow well, glittering,
Silent, savage, weird.

Mr. Morris sent a group of these to the student literary magazine at his University, but they were not printed. Apparently, they smelled of the machine. Or perhaps the lack of quality was plainly perceptible to a student poetry editor.

The foregoing examples illustrate the difficulties of computer poetry generation on the technical level of versification (rhyme, meter). These difficulties, however, can be dealt with by means of suitable correction routines applied after the sentences have been generated. Moreover, they are not fundamental to the problem in that neither rhyme nor meter is an essential constituent of poetry. The problems of imagery (e.g., metaphorical consistency) raised by this process are in a different class. Unlike versification features, which are linguistic at the phonological level, imagery features are linguistic at the semantic level or extra-linguistic, in that they cannot be detected without reference to the world of things. Only linguistic features, those which can be specified by rules applicable to the world of symbols, can be taken into account in poetry generation algorithms. Consequently, such imagery problems are of minor interest for poetry generation at the present stage of development of semantic theory.

More interesting, both from a theoretical and a practical viewpoint, is the question of randomness or uncertainty: how to produce well-formed sentence strings whose syntactic pattern has not been previously determined by the algorithm. The basic difficulty consists in introducing enough uncertainty to insure variety in the resultant output without violating the constraints

that govern well-formed sentences. Though this is primarily a practical problem, it touches on theoretical considerations of syntax primarily in the taxonomy of word-classes and the placement of constructions. When, for example, a simple (static) sentence pattern is established and generation merely consists of inserting vocabulary items from various lists into syntactical slots (as in example on p. 3, above), the resulting sentences are likely to be well-formed, if care has been taken to confine the membership of each list to items with distributional parity. When, on the other hand, the principle of true recursion is given free play, when that is any word may be replaced by a larger construction (adverb by adverbial clause, adjective by relative clause), the possibility of producing ill-formed sentences exists. Between these extremes there is a middle ground involving the use of iteration and option which I have exploited in my current work.

The basis of my RETURNER programs is a poem entitled "Return" by Mrs. Alberta Turner (Reference 12, and see Appendix). This poem has served as the source of vocabulary items, both individual words and some phrases, in the preparation of the programs. Of the five versions of this program that have been developed, the first was merely a trial and the fourth and fifth have been abandoned as unpromising. Consideration is therefore limited to Versions Two and Three.

RETURNER TWO is based on a pattern consisting of subject and verb, with optional modifiers and complements, and an optional conjunction at the end which provides the opportunity of iteration. Options are decided by drawing a random digit (1 is success, 0 is failure). The subject having been selected, the first option is for a post-modifier. Then the verb is selected and followed by an optional complement, which may be a direct object or an adverbial. The third and final option is for a coordinating conjunction, in which case

the process begins again. The following selections from the output illustrate the possibilities from the simplest to the most elaborate*:

61. Deer / planted (subject-verb)

87. Hemlocks / saw / a thin curd (subject-verb-object)

12. The snow / planted / again (subject-verb-adverbial)

117. Tracks / around it / shelled (subject-modifier-verb)

125. Apple twigs / from salad to salad / saw / the sheep (subject-modifier-verb-object)

50. The pond / through the willows / planted / yesterday (subject-modifier-verb-adverbial)

93. Deer / shelled / yet / my dog / wouldn't take (subject-verb-conjunction-subject-verb)

109. The pond / shelled / yet / the porch post / wouldn't take / and / the snow / saw / the sheep (subject-verb-conjunction-subject-verb-conjunction-subject-verb-object).

The attractive unpredictability of the output of this program suggested that a more elaborate design would yield more interesting results. The consequence, RETURNER THREE, is based on a tagmemic matrix of six slots: a front adverbial, the subject, a middle adverbial, the verb system (including complement), an end adverbial, and a terminal conjunction (in the case that iteration is selected). Each slot leads to the selection of a construction. The subject, for example, may be a pronoun, a noun, or a complex nominal. The adverbials and the conjunction include a zero option. The choice of con-

---

* Slashes show boundaries between constructions.

struction then leads to a list from which the particular lexical unit is chosen. To illustrate the process, the following list of steps is involved in the generation of the verb system:

I. The 'V' slot is selected from the tagmemic matrix.

II. A choice is made among copulative, transitive and intransitive verbs.

III A. If the copulative verb option has been selected, a verb is chosen from the appropriate list and the route leads to the selection of a noun or adjective complement, the choice being decided by an odd/even test of the number of words in the string.

1. The adjective is chosen from an adjective list.

2. The noun complement is chosen from the noun list, one of three lists available for the choice of subject.

B. If an intransitive verb is selected, a verb is chosen from the appropriate list and an optional adverbial complement is added.

C. If a transitive verb is selected, a verb is chosen from the list and a direct object is added, either

1. a nominal construction or

2. a pronoun in objective case.

IV. The routing then proceeds to the next slot in the matrix.

After the string has been generated but before the iteration (if any) occurs, the string is subjected to a set of adjustment routines to insure grammatical agreement and concord:

1. In the case of the verb be, suitable forms are substituted in accordance with the person of the subject.

2. If the presence of a leading adverbial (yesterday) signals a past tense, a test is made for verb regularity:

a. if the verb is irregular, the suitable past form is substituted from a list,

b. if the verb is regular, the past morpheme is attached, subject to the necessity of orthographic adjustment (e.g., doubled final consonant).

3. If the leading adverbial has a future sense, the auxiliary will is added before the base form, except that in the case of be the base form is substituted for the inflected form already in place.

4. If neither tense adjustment is required and the verb is not be, a test is made for the possibility that the person of the subject requires the singular verb morpheme (-s) and this is added as needed, subject to phonological rules.

The final option is rhetorical. If the odd/even test so dictates, a subject - verb inversion is applied, or a verb-object/complement inversion, except that the collision of pronouns in inverted position is avoided as unidiomatic. After iteration is complete and a clean-up routine is performed which removes signals, markers, diacritics and other symbols internal to the program, the generated string is set out as a stanza by a function that makes each line as long in words as the first word of each line has letters.

The following stanzas have been generated by this program, RETURNER THREE-C:

2. In the
Morning melons often fall and tomorrow separate
Blankets will bring me through the willows.

7. They turn locusts today
Yet at home
Crowbars never stagger in the morning.

9. Tomorrow a thin curd often will shell her
In the
Morning.

16. Often was he nearly gray
Yesterday.

19. We again
Will appear nearly round
Tomorrow.

27. Yesterday hemlocks planted all the apples at home yet
They often turn crowbars
Today.

48. Turn the crowbars today
So through
The willows my
Dog again staggers
At home
And carefully separate
Blankets knead me in the morning.

Despite a tiresome lack of variety, both in syntax and in vocabulary, these synthetic stanzas have an unmistakable "poetic" quality and a family relation to the source poem. It would be too simple to ascribe this effect merely to the presence of the same words in "Return" as in RETURNER. Nor do we believe any longer in such a thing as a poetic diction, unless a diction is poetic which collocates crowbars with kittens, dogs and staggering. The unexpected collocations of poets like Dylan Thomas have perhaps oriented us to a different conception of poetic texture or a different sound of the poet's voice.

The unquestionably primitive poetic output of RETURNER can be brought closer to the source poem by adding linguistic refinements to the program design. The next version will include more of the vocabulary of the original, will attempt to incorporate syntactic algorithms for subordination, possibly a question transformation and a less random stanzaic formula. With enough refinements, RETURNER

should be able to generate "Return," not only in its original form but in the alternative forms which Mrs. Turner presumably wrote and discarded. But this is not an easy task. The difficulties so far overcome (for instance, the priority of decisions applied in the adjustment routines) suggest that one can more easily write poetry with pencil and paper than with a computer. But my experience with this humbling process has persuaded me that it is a good way to learn about the design of poetry and the reading of it.

A consideration of the problems involved in generating computer poetry alerts us to the curious behavior of familiar words in unfamiliar combinations or contexts. We perceive how readily we accept metaphor as an alternative to calling a sentence nonsensical. We tend, that is, always to try to interpret an utterance by making whatever concessions are necessary on the assumption that the writer had something in mind of which the utterance is the sign. Of course, this is inappropriate when the speaker is a computer. The consequence seems to be the demolition of the critical axiom that the poem is sufficient. If we are not to waste our time in vain interpretation we must now ask a new question before beginning an exegesis: Who or what wrote this poem? The problem, however, will not arise in a serious form until computer poetry becomes somewhat better than it has been until now.

APPENDIX

Hemlocks are nearly round,
Deer paw the pond,
My dog squirts the porch post.

Last night the snow wouldn't take tracks,
But apple twigs are cut
Higher than porcupines.

Yesterday I saw the weathercock
Through the willow;
Today the cock is gone.

Holding my bowl,
I step carefully
From salad to salad.

The swamp has shores again
And the quicksand grass.

Have I planted crowbars under my porch
And chisels under my rocker?

No crumbs fall from the agate pebble,
But around it acres of sand are also red.

Melons crack,
Locusts have shelled my sisters on the porch,
The collie's tongue sticks.

The kittens hiss, at the milk pan,
And knead separate blankets.

This morning all the apples ringed the tree
So close the boy turned his ankle
And rabbits staggered.

The jack o'lantern's soft now
And nearly gray.

Mica silvers the sheep.
When my child brings his paper star,
Will it glitter? It's not silver paper.

Hemlocks are nearly round,
Mice run under the snow,
Sunrise reddens a thin curd.

BIBLIOGRAPHY

1. Richard W. Bailey, "Automating Poetry," Computers and Automation, XIX (April 1970), 10, 13.

2. Jean A. Baudot, La machine à écrire (Montreal, 1964).

3. "Computer Poems and Texts," Cybernetic Serendipity (London, 1968), pp. 53-62 [articles by Marc Adrian; Robin McKinnon Wood and Margaret Masterman; Nanni Palestrini; Alison Knowles and James Tenney; Edwin Morgan; Jean A. Baudot; and E. Mendoza].

4. Wilbur Cross, "Machine Miltons," The New York Times Magazine, Dec. 4, 1966, pp. 59, 62, 64.

5. "Electronic Poetry," Electronic Age, XXII, 3 (Summer 1963), 30-31.

6. Samuel R. Levin, "On Automatic Production of Poetic Sequences," Texas Studies in Literature and Language, V, 1 (Spring 1963), 138-146.

7. J. H. Martin, Correspondence in Times Literary Supplement, March 19, 1964, p. 235.

8. Margaret Masterman, "The Use of Computer to Make Semantic Toy Models of Language," Times Literary Supplement, August 6, 1964, pp. 690-91.

9. Louis T. Milic, "Computer Programs and the Heroic Couplet," Think, May-June 1968, pp. 28-31.

10. John Morris, "How to Write Poems with a Computer," Michigan Quarterly Review, VI (January 1967), 17-20.

11. "Poetry, Prose and the Machine," Freeing the Mind (London, [1962]), pp. 45-49.

12. Alberta T. Turner, "Return," The Midwest Quarterly, IX, 2 (Winter 1968), 168-169.

13. Jon Wheatley, "The Computer as Poet," Queen's Quarterly, LXXII (Spring 1965), 105-120.

14. Victor H. Yngve, "Random Generation of English Sentences," 1961 International Conference of Machine Translation and Applied Language Analysis (London, 1962), Vol. I, pp. 66-80.

15. Margaret Masterman and Robin McKinnon Wood, "The Poet and the Computer," Times Literary Supplement, June 18, 1970, pp. 667-668.